ORC 64-27
OCTOBER 1964

SOME INEQUALITIES GOVERNING OPTIMUM CODE

by

T. C. Hu

OPERATIONS RESEARCH CENTER
INSTITUTE OF ENGINEERING RESEARCH

UNIVERSITY OF CALIFORNIA - BERKELEY

SOME INEQUALITIES GOVERNING OPTIMUM CODE

by

T. C. Hu
Operations Research Center
University of California, Berkeley

23 October 1964 ORC 64-27(RR)

This research has been partially supported by the Office of Naval Research under Contract Nonr-222(83) and the National Science Foundation under Contract GP-2633 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

ACKNOWLEDGEMENT

The author is indebted to Dr. R.M. Karp for stimulating discussions.

Some Inequalities Governing Optimum Code

Let an information source be given which generates messages consisting of sequences of letters $\{X_1, X_2, \ldots, X_n\}$. Each letter X_i occurs with probability p_i. In practice, for example, X_i may be the alphabet and p_i the frequencies of usage in English. These letters X_i are to be encoded for transmission over a communication channel admitting the symbols α and β.

In the present paper, we consider prefix code. In 1952, using an elegant combinatorial approach, Hoffman [1] obtained an optimum prefix code for the case the symbols α and β cost the same. Later Karp [2] used integer programming to obtain optimum prefix code with symbols of different costs. Here, we use combinatorial argument to study the case where the "α" costs d dollars and the "β" costs $d + 1$ dollars where d is a positive integer. This case will reduce to the Hoffman's case when d becomes infinite, and for $d = 1$, approximate the dot-dash case of common usage.

A prefix code may be described by a tree as shown in Figure 1.

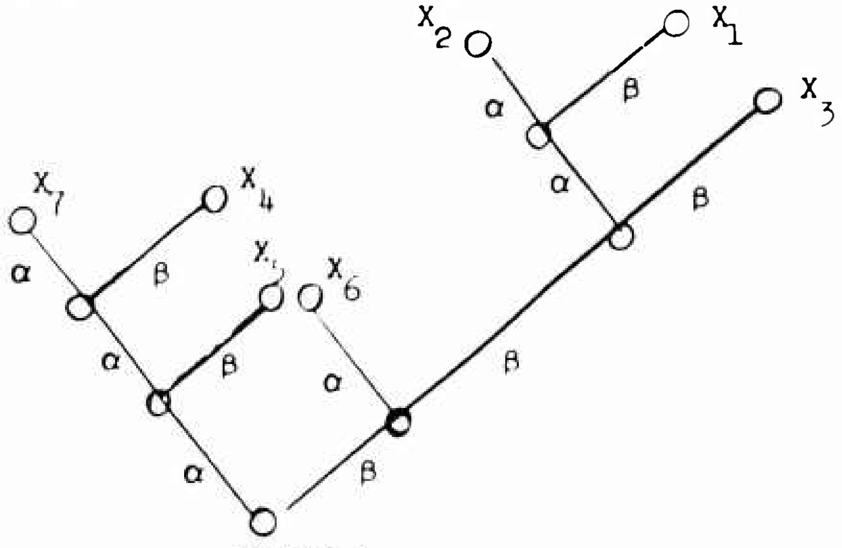

Figure 1

Each terminal node is associated with a letter X_i . The branches leaving each node are labeled with names of distinct symbols α and β , and the code word C_i of each X_i is found by listing in the order the labels of the branches leaving the root of the tree to the terminal node associated with X_i . Thus, in Figure 1, the code word for X_5 is $\alpha\beta$ and the code word for X_6 is $\beta\alpha$. The length ℓ_i of a letter X_i is the sums of α's and β's used in the code word. Thus, X_5 and X_6 have the same length and X_1 is of length $\alpha + 3\beta$. The length of X_i is a direct measure of its cost. Once a tree, such as in Figure 1, is given as a prefix code, the cost of the code is given by

(1) $$\sum_i p_i \ell_i \qquad (i=1,2,\ldots,n) \quad .$$

The problem of constructing a optimum prefix code is, with given p_i, to find a tree such that (1) is minimum. Assume X_i are indexed so that

(2) $$p_n \geq p_{n-1} \geq \cdots \geq p_2 \geq p_1 \quad .$$

Then for an optimum code, it is necessary

(3) $$\ell_n \leq \ell_{n-1} \leq \cdots \leq \ell_2 \leq \ell_1 \quad .$$

If (2) (3) are not satisfied, we could interchange the code words and reduce the value of (1).

Let us define $\bar{\ell}_i$ of a letter X_i to be the length of X_i minus the last symbols in the code word representing X_i . In Figure 1, for example, if we discount the last symbol α representing X_2 , its $\bar{\ell}_2$ is $\alpha + 2\beta$. Similarily the $\bar{\ell}_3$ of X_3 is 2β and $\bar{\ell}_6$ of X_6 is β . Since the cost of α is an integer and the cost of β is also an integer, the $\bar{\ell}_i$ of X_i and $\bar{\ell}_j$ of X_j are also integers. And if

$$(4) \qquad \bar{\ell}_i < \bar{\ell}_j$$

then they differ by at least 1, i.e., $\bar{\ell}_i + 1 \leq \bar{\ell}_j$.

Assume the last symbol of X_i is β and the last symbol of X_j is α . As (4) implies $\bar{\ell}_i + d + 1 \leq \bar{\ell}_j + d$, we have (It is clear that (5) is true if the last symbol of X_i is α and that of X_j is β or the case both X_i and X_j have the same last symbol),

$$(5) \qquad \ell_i \leq \ell_j \quad .$$

Therefore, for an optimum code, (4) implies (5), and (5) implies $p_i \geq p_j$.

Assume now that the optimum prefix code, there are $2m$ letters with longest $\bar{\ell}$. Then let the $2m$ letters with less probabilities be $X_{2m}, X_{2m-1}, \ldots, X_1$ with $p_{2m} \geq p_{2m-1} \geq \cdots \geq p_1$. Then obviously for an optimum prefix code, these $2m$ letters are the terminal nodes of the m longest $\bar{\ell}$. Disregard the rest of the tree structure representing the optimum code for a moment; we can symbolically represent the part of the tree as in Figure 2.

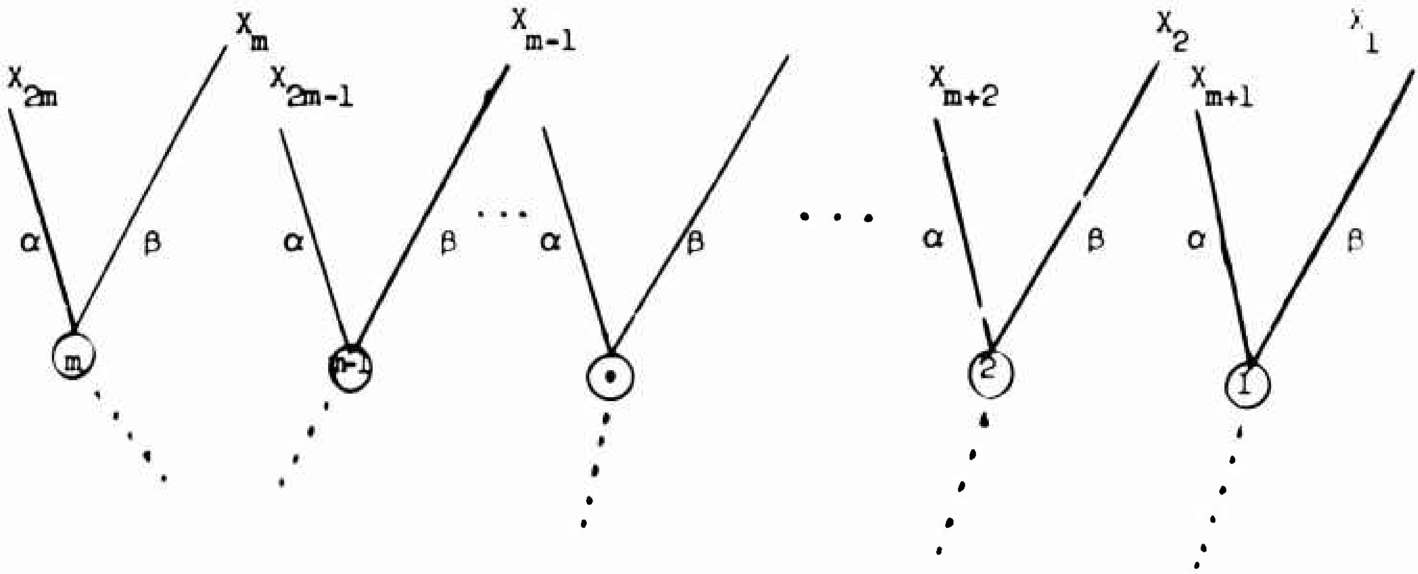

Figure 2

Note the arrangement in Figure 2 is not unique. Any assignment of X_{2m}, $X_{2m-1}, \ldots, X_{m+1}$ to the α branches and any assignment of X_m, $X_{m-1}, \ldots, X_1$ to the β branches will have the same total cost. We shall study several inequalities which permit us to simplify the construction of a prefix optimum code. First, if

(6) $$p_{2m} \geq p_{m+1} + p_1 \qquad ,$$

then we can rearrange Figure 2 into Figure 3 without changing the rest part of the tree and not increase the total cost.

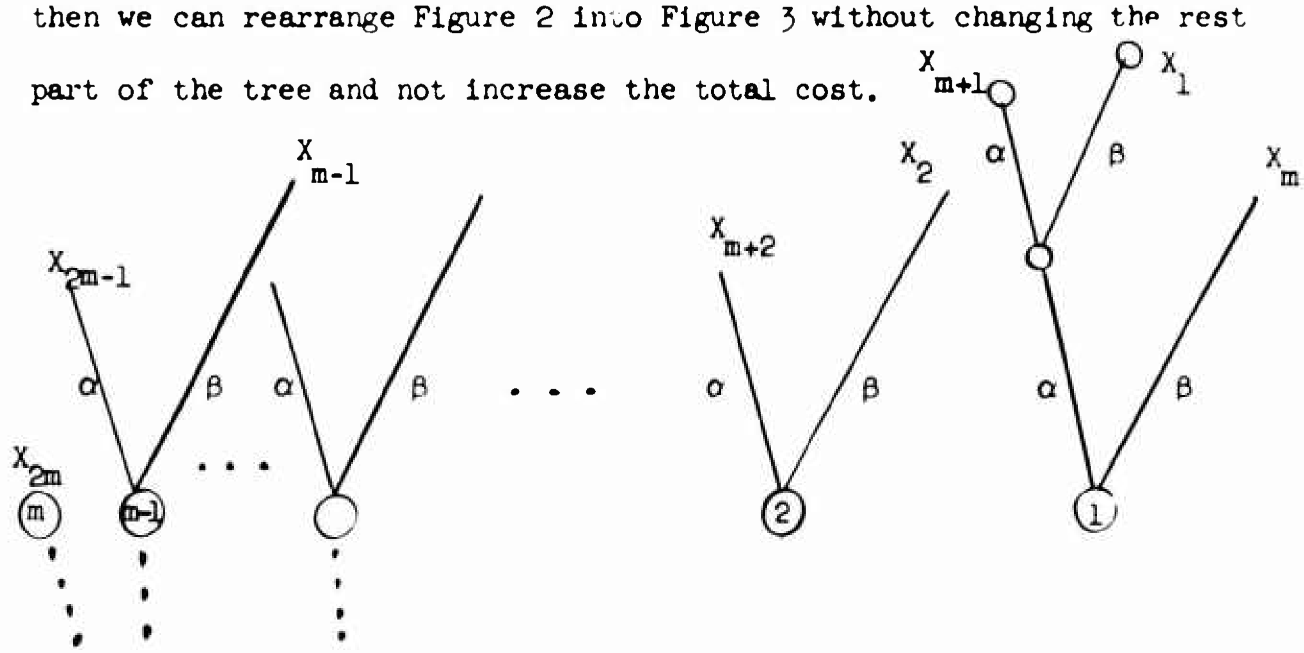

Figure 3

This is because changing from Figure 2 to Figure 3, the decrease in cost is $p_{2m} \cdot d$ and the increase in cost in $(p_{m+1} + p_1)d$. Therefore, if (6) is true, then there exists an optimum prefix code in which the maximum number of longest $\bar{\ell}$ is less than m .

In particular, for $m = 2$ then (6) becomes

(7) $$p_4 \geq p_3 + p_1$$

and there is only one $\bar{\ell}$ of longest length. For this optimum code, on that longest $\bar{\ell}$, the two terminal nodes associated with it will be X_2 and X_1 . This means X_2 and X_1 will have the same code word except the last symbol where X_2 has α and X_1 has β . In constructing an optimum prefix code, we can treat X_2 and X_1 as one letter with probability equals to the sum of p_2 and p_1 as done by Hoffman [1].

Secondly, if

(8) $$d.p_{2m} \geq (d + 1)p_1 + d\, p_2 \quad ,$$

then we can change Figure 2 into Figure 4 below without increasing the total cost.

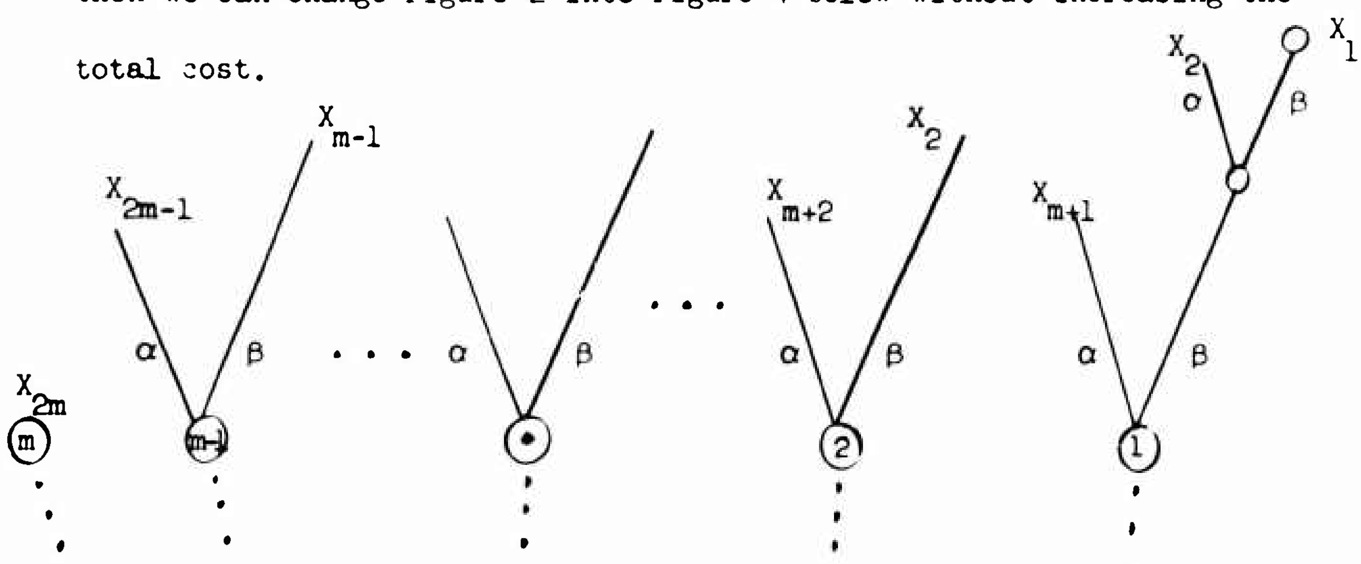

Figure 4

This is because in changing from Figure 2 to Figure 4 the decrease in cost is $d.p_{2m}$ and the increase in cost is $dp_2 + (d + 1)p_1$.

This means if (8) is satisfied, then there exists an optimum tree in which the maximum number of longest $\bar{\ell}$ is less than m. In particular for $m = 2$, then (8) becomes

$$(9) \qquad dp_4 \geq (d + 1)p_1 + dp_2$$

and there is only one $\bar{\ell}$ of longest length. Again we can associate X_2 and X_1 with this $\bar{\ell}$ and hence reduce the total number of letters by one. Note that if (7) is satisfied so will be (9) so this really does not give us any new inequality. But if (6) is satisfied, (8) may not be. Third, if

$$(10) \qquad dp_{2m} \geq p_{m+1} + (d-1)p_2 + dp_1$$

then we can change Figure 2 to Figure 5 without increasing the total cost

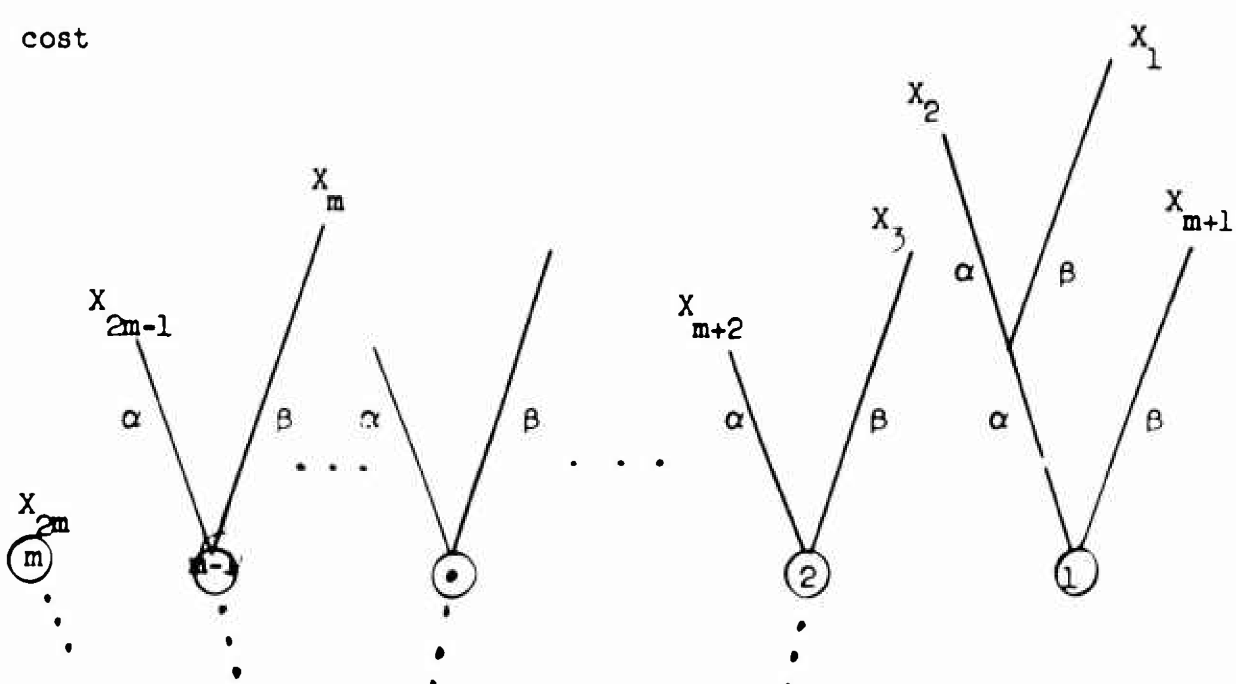

Figure 5

This is because the right hand side of (10) represent the increase in cost, and the left hand side of (10) represent the decrease in cost in changing from Figure 2 to Figure 5. For $m = 2$, (10) will reduce to (7). Therefore if (7) is satisfied, we can combine X_2 and X_1 and regard them as a single letter with probability equal to the sum of p_2 and p_1. If in the newly created $n - 1$ letters X'_{n-1}, $X'_{n-2}, \ldots, X'_1$ we also have $p'_4 \geq p'_2 + p'_1$, we can again combine X'_2 and X'_1 into one letter. This process can be continued until (7) is not true. Note that if ( ) is not satisfied, the number of longest $\bar{\ell}$ may still be one.

Let $m = 2$ for (6), (8) and (10), we have the following inequalities

(11) $$p_6 \geq p_4 + p_1$$

(12) $$dp_6 \geq dp_2 + (d + 1)p_1$$

(13) $$dp_6 \geq p_3 + (d + 1)p_2 + dp_1$$

If anyone of (11), (12), or (13) is satisfied, then the maximum number of longest $\bar{\ell}$ is at most 2. If we knew that the number of longest $\bar{\ell}$ is exactly 2, then we can combine X_1 and X_3 into one letter and X_2 and X_4 into one letter (see Figure 6). In order to be able to combine X_1 and X_3 and also X_2 and X_4, we study in more detail the part of the tree with terminal nodes X_2 and X_1. If we study one more level of the part of the tree containing X_2 and X_1 and assume that there is only one longest $\bar{\ell}$, there are only five possible configurations as shown in Figure 6, 7, 8, 9, and 10.

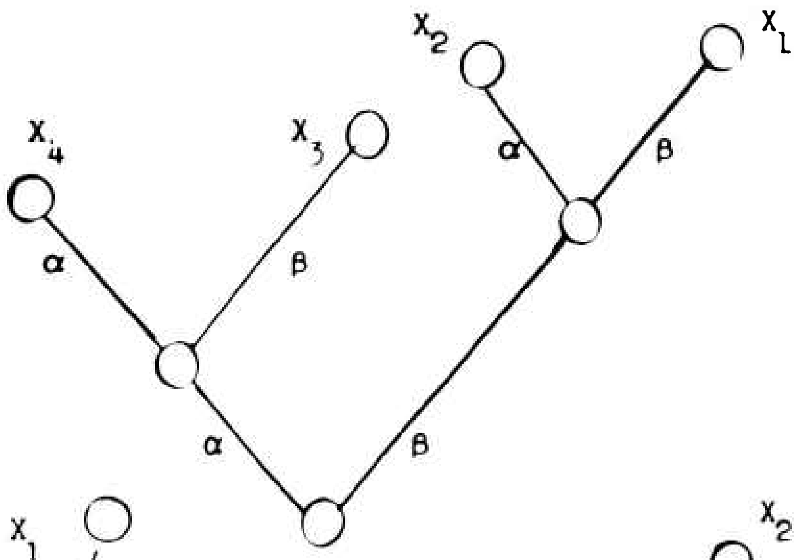

Figure 6

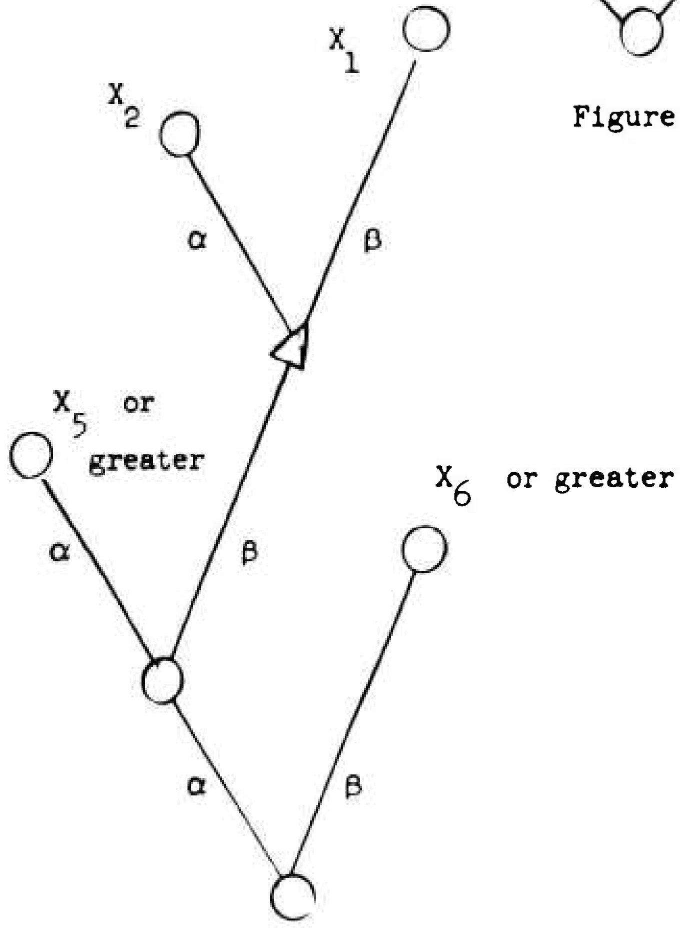

Figure 7

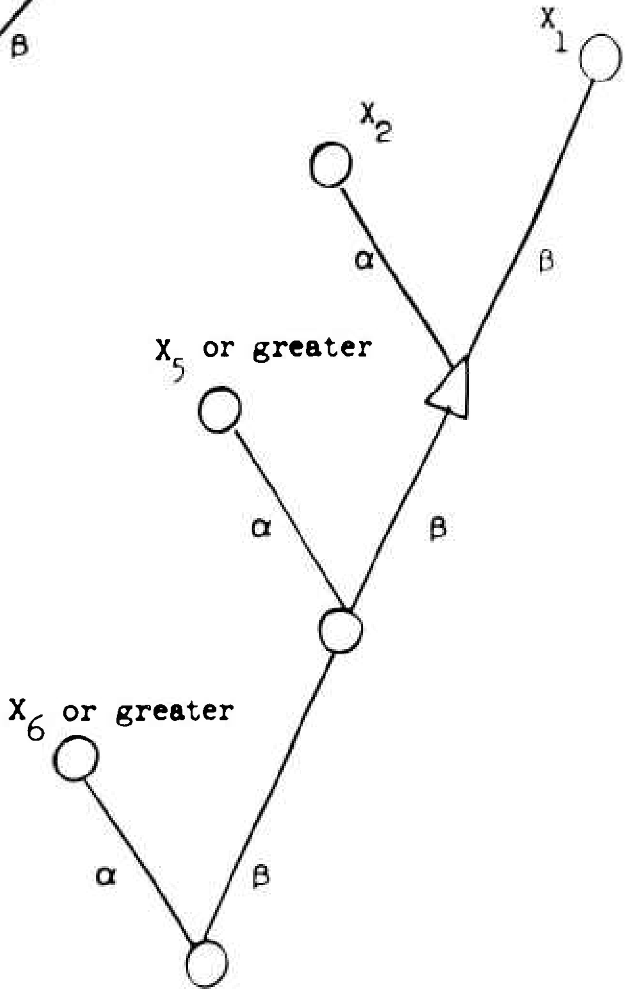

Figure 8

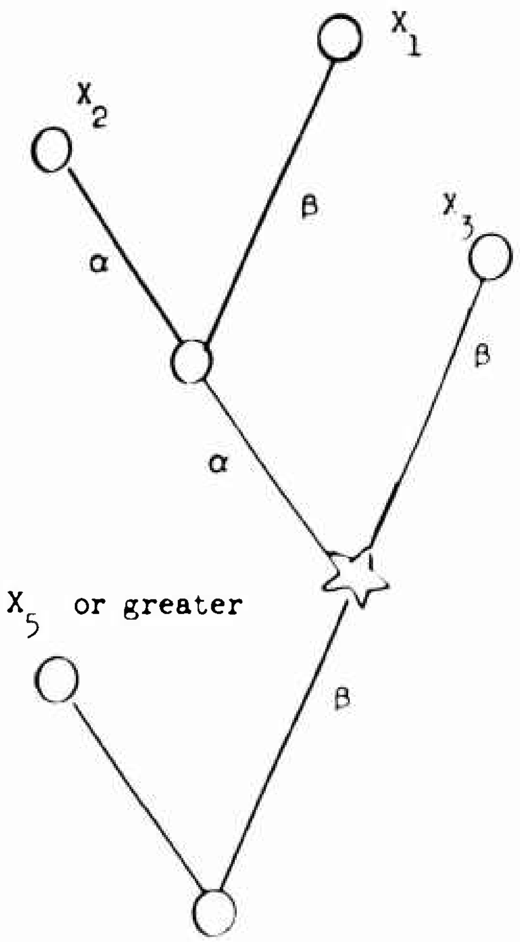

Figure 9

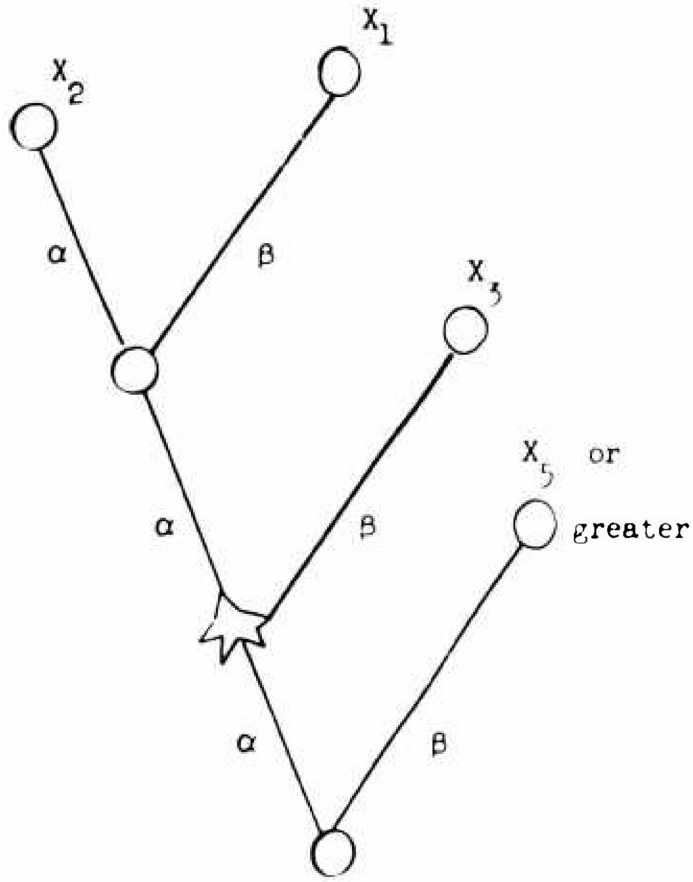

Figure 10

In Figure 6, we can interchange the code words associated with X_3 and X_2 without changing the total cost, i.e. we can still combine X_1 with X_3 and X_2 with X_4 ,even $\bar{\ell}$ is one. In Figure 7 or Figure 8, we have written X_5 or greater, X_6 or greater. This is because we have assumed that (7) is not true. . For an optimum code, we cannot assign a letter X_3 or X_4 which has less probability than $p_2 + p_1$ with a length shorter than $\bar{\ell}$ of X_1 and X_2 so that the terminal node associated with a certain branch must be X_5 or letters of greater probabilities.

In Figure 7 or Figure 8, X_3 is not in the figure, but the last symbol of the code word for X_3 must be β as X_3 is the letter with the smallest probability not in the Figure 7 or 8. The last symbol of the code word for X_4 may be α or β ; we shall assume it to be α in order to be on the safe side.

Then we can transfer the X_3 and X_4 into the part of the tree containing X_1 and X_2 in Figure 7 or 8 and make it look like Figure 11.

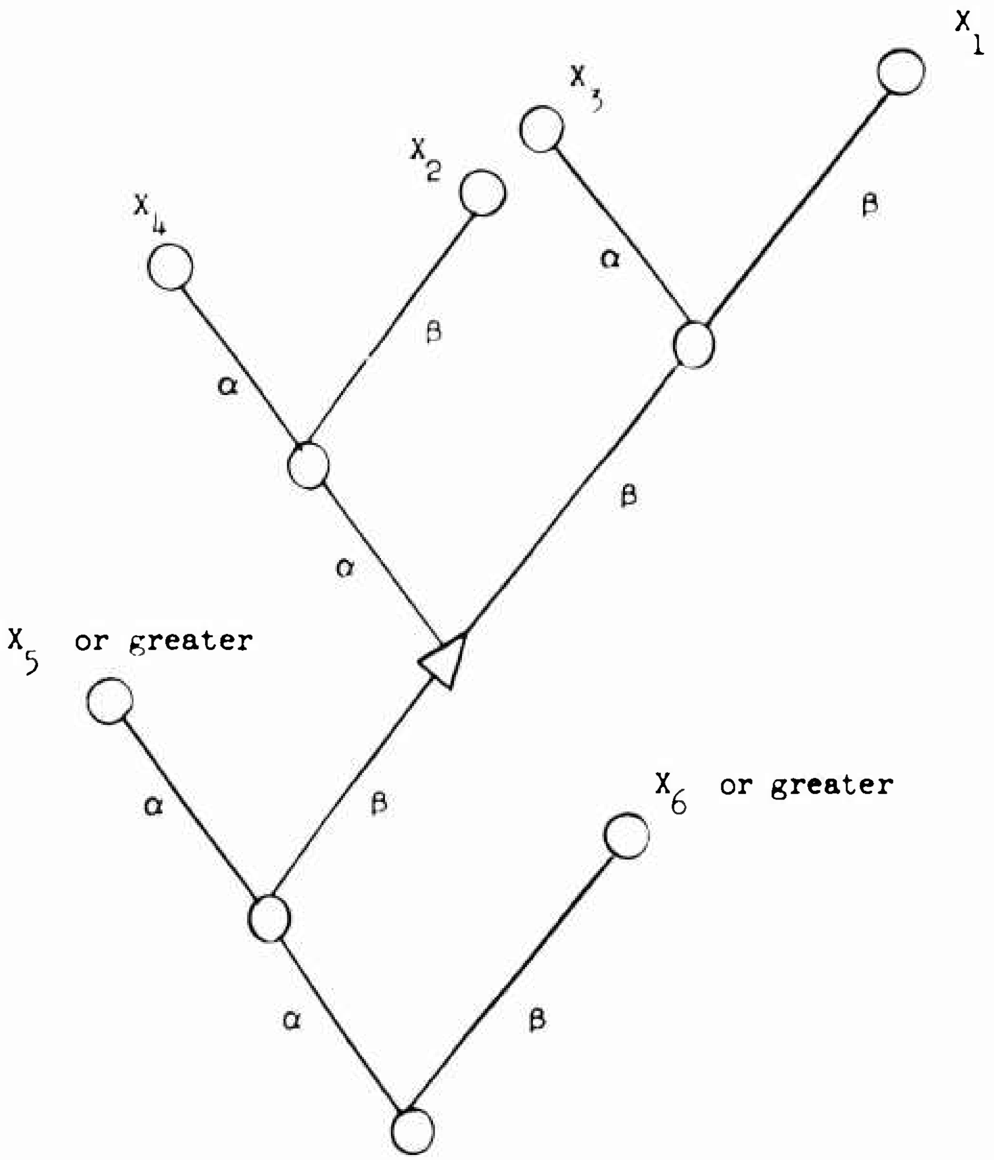

Figure 11

The letters that originally combine with X_3 and X_4 can then reduce their code word by one symbol, say α , to be on the safe side. These letters must have probabilites p_5 and p_6 or greater. So

In changing from Figure 7 or Figure 8 to Figure 11, the reduction in cost is at least $(p_5 + p_6)d$, where the total increase in cost is at least $(p_1 + p_2)d + (p_3 + p_4)d$. Therefore if

$$(p_5 + p_6)d \; [illegible] \; (p_3 + [illegible])d - (p_1 + p_2)(d - [illegible]) \tag{14}$$

then we can change Figure 7 or Figure 8 into Figure 11 with no increase in cost. Note in Figure 11 we do combine X_1 with X_3 and X_2 with X_4 . Consider Figure 9 and Figure 10. As X_4 is the letter with smallest probability not shown in the Figure 9 and 10, the last symbol of X_4 must be β . The letters that combine with X_4 must have a last symbol of α and a probability of p_5 or greater. So we can change Figure 9 and Figure 10 into Figure 12.

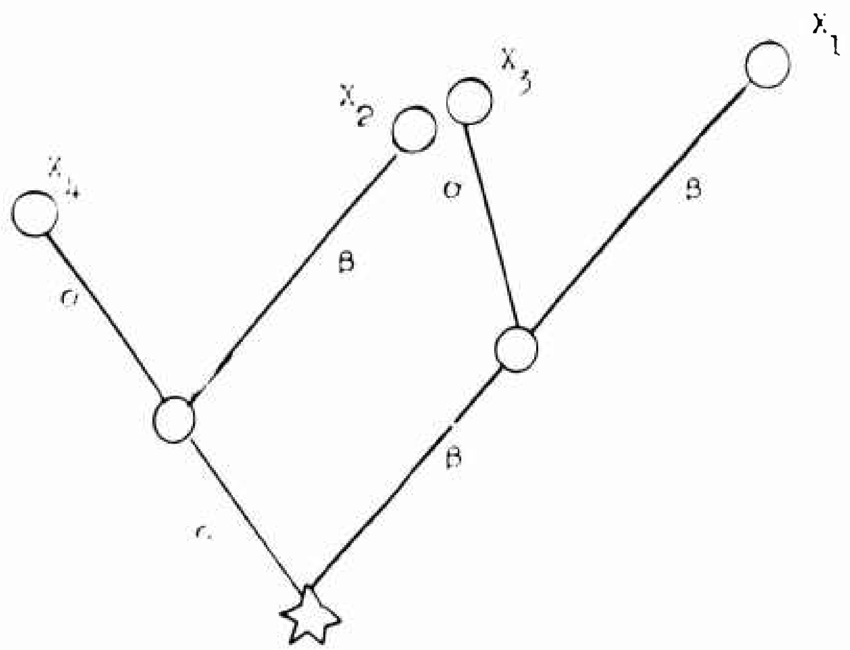

Figure 12

The total increase in cost is $\quad p_1(2\beta - \alpha - \beta) = p_1$

$$p_2(\alpha + \beta - \alpha - \alpha) = p_2$$

$$p_3\,\alpha \qquad = p_3 d$$

$$p_4(2\alpha - \beta) \qquad = p_4(d-1)$$

Total decrease in cost is $\quad p_5.\alpha \qquad = p_5.d$

If

$$\text{(15)} \qquad dp_5 \geq (d-1)p_4 + dp_3 + (p_2 + p_1) \quad ,$$

then we can change Figure 9 or Figure 10 into Figure 12 in which X_1 combine with X_3 and X_2 with X_4 .

If any one of (11), (12), or (13) is satisfied then the maximum number of $\bar{\ell}$ is at most two. If it is two, then we can combine X_1 and X_3 and X_2 with X_4 and reduce the number of letters. If there is only one $\bar{\ell}$, then there are only five figures possible as shown in Figure 6,7,8,9, and 10. So if (14) and (15) are satisfied, we still can combine X_1 with X_3 and X_2 with X_4 , hence reduce the number of letters.

In applying the inequality (7) to the example $d = 1$ in the paper by Karp [2], the number of letters is immediately reduced by 5 .

REFERENCE

[1] D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," Proc. I.R.E., Vol. 40, pp. 1098-1101, Sept. 1952.

[2] R. M. Karp, Minimum-Redundancy Coding for the Discrete Noiseless Channel," I.R.E. Trans. on Information Theory, Vol. IT-7, Number 1, January 1961.